Scientific production setup

Nasser Alkmim

<2021-06-03 Thu>

Contents

Emacs is by far the most advanced LaTeX editor that I have used. Why? It makes things easier and more enjoyable. The main goal is reduce computational friction between what needs to be done and your tools.

Under the hood there are couple of packages that enable that.

1 AUCTeX

It is a major mode for `.tex` files. Is an environment for writing input files for TeX documents.

With AUCTeX is easier to:

1. insert macros

2 RefTeX

Useful to create and reference labels.

3 Inserting macros

Type \ and you are prompt to choose: `\sqrt{}, \int, \dfrac{}{}`.

A *completion framework* (e.g. Vertico, Selectrum, Ivy, Helm) would enhance even more the functionality by showing the possibilities and narrowing it.

4 Fonts

4.1 Newtx math libertine

$$\boldsymbol{\varepsilon} = \boldsymbol{\epsilon}^{e} + \boldsymbol{\epsilon}^{p}$$

$$\boldsymbol{\sigma} = \mathbf{D}^{e} : \boldsymbol{\epsilon}^{e}$$

4.2 Newtx math utopia

$$\boldsymbol{\epsilon} = \boldsymbol{\varepsilon}^{e} + \boldsymbol{\varepsilon}^{p}$$

$$\boldsymbol{\sigma} = \mathbf{D}^{e} : \boldsymbol{\varepsilon}^{e}$$

5 Outline

5.1 Workflow to narrow to focus

I use `outline-hide-other` to focus on a specific heading while still shows the context.

to show the sibling of a heading I use `outline-show-children` on the parent heading (jump with `outline-up-heading`).

If I want to see just the headings structure I use `outlie-hide-leaves`.

6 Zotero

Workflow:

1. copy DOI and import in Zotero.
2. Better Bibtex automatic update `.bib` file.
   (a) bibfile stays at `~/.bibliography.bib`.
3. Zotfile automatic rename the pdf, move it to a specific folder and creates a link for Zotero to find it.
   (a) pdfs stays in a cloud folder `~/SeaDrive/<pdfs>`

7 Matplotlib plots

7.1 Quote

> What is to be sought in designs for the display of information is the clear portrayal of complexity. Not the complication of the simple; rather the task f the designer is to give visual access to the subtle and the difficult – that is, the revelation of the complex.
>
> – Edward R. Tufte

7.2 Finding size of latex text

This will print the width of the page element.

The `textwidth` is the width of text area and it is dependent on the page margin. `linewidth` depends on the environment. `columnwidth` is the column of text and depends on the number of columns.

```
\the\textwidth
\the\linewidth
\the\columnwidth
```

7.3 Matplotlib size and LaTeX size (pt)

Matplotlib default uses inches.

Latex uses points (pt), 72.27 pt = 1 in.

7.4 Ideal plot size

If the nature of the data does not suggests the shape of the plot, then the length should be larger than the height. The reasons are (Tufte, Edward R., 2013):

1. analogy to the horizon, which I like because of the "poetic" argument.
2. easier to label.
3. emphasis on causal influence. This is my favorite reason because a longer horizontal length allows a more detailed exploration of the relation.

The proportion that most prefer is close to the golden ratio. I use the height as 60% of the length.

7.5 The case against `.pgf`

Back in 2017 I tried to use it, but soon discovered that is impractical for any substantial work. It takes too long to compile. Now I use just use `.pdf` without scaling the plot.

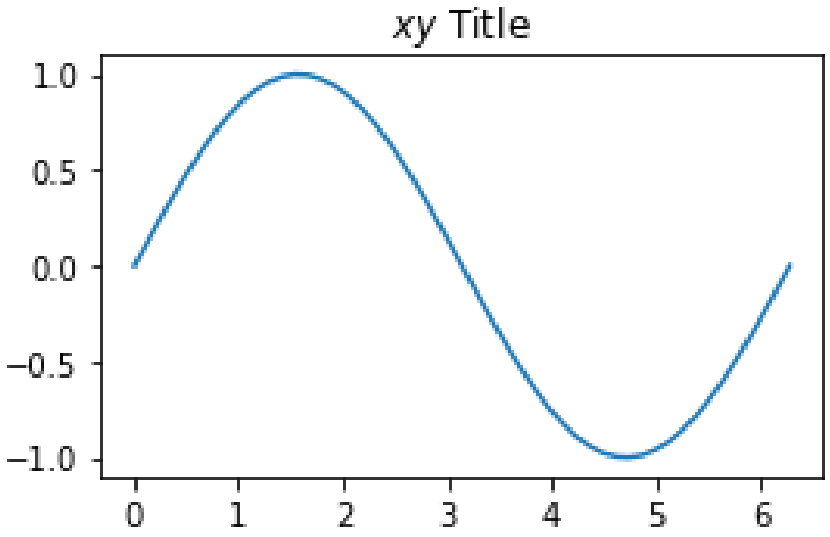
7.6 Testing

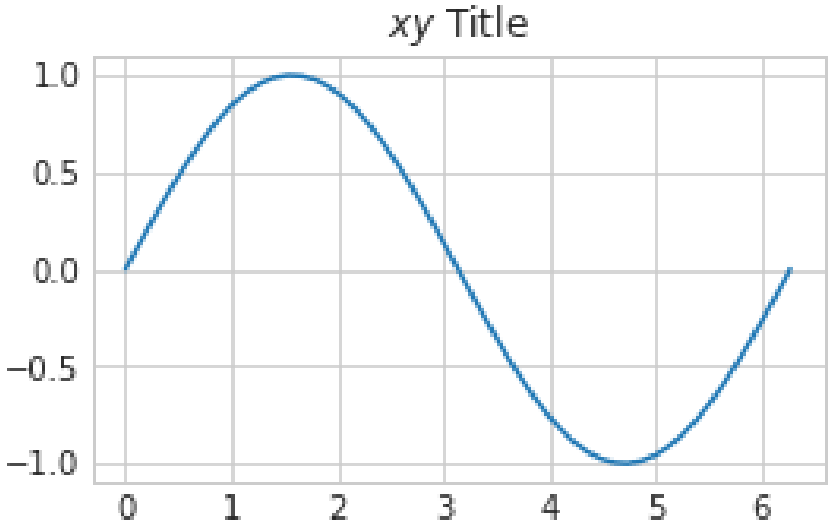
Remarks:

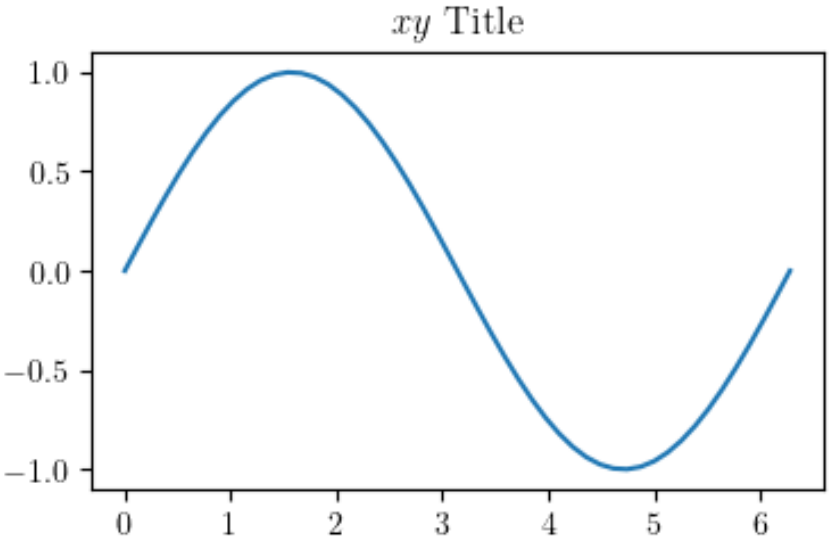
1. when `usetex` is used, the plot size changes even though the figure was created with the same size.
2. if no font is specified, "computer modern" is used.

```
import numpy as np
import matplotlib.pyplot as plt

size_pt = 300
size_in = size_pt / 72.27
figsize = [size_in, size_in * 0.6]

x = np.linspace(0, 2 * np.pi)
y = np.sin(x)

fig, ax = plt.subplots(figsize=figsize)
```

```
ax.plot(x, y)
ax.set_title('$xy$ Title')

plt.style.use('seaborn-whitegrid')
fig, ax = plt.subplots(figsize=figsize)
ax.plot(x, y)
ax.set_title('$xy$ Title')

plt.style.use('default')
plt.rcParams.update({'text.usetex': True, 'font.family': 'serif'})
fig, ax = plt.subplots(figsize=figsize)
ax.plot(x, y)
ax.set_title('$xy$ Title')
```







7.7 Why use a centralized configuration system

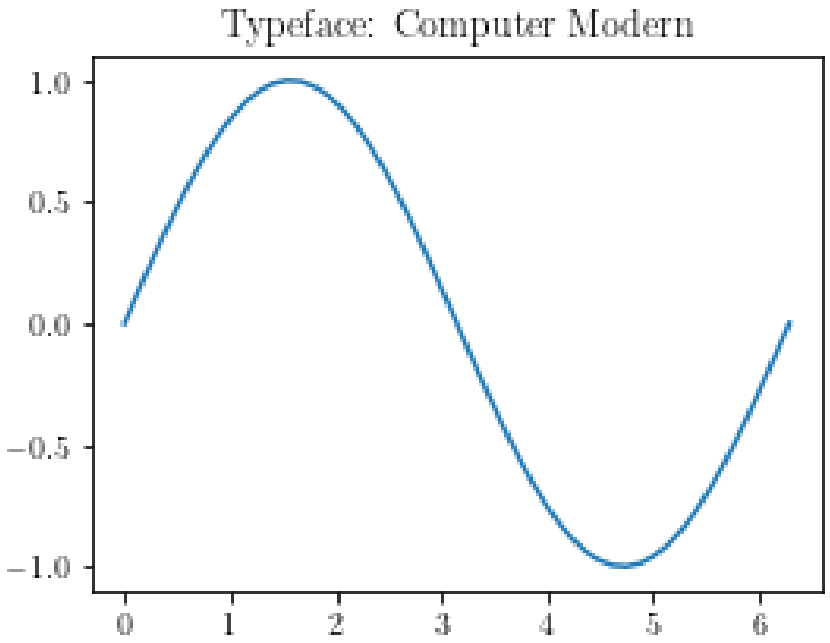
A central place where the plot style is configure is ideal because it enforces consistency and it is easier to change in a single place.

A use a simple package that I install locally with `pip install -e .`.

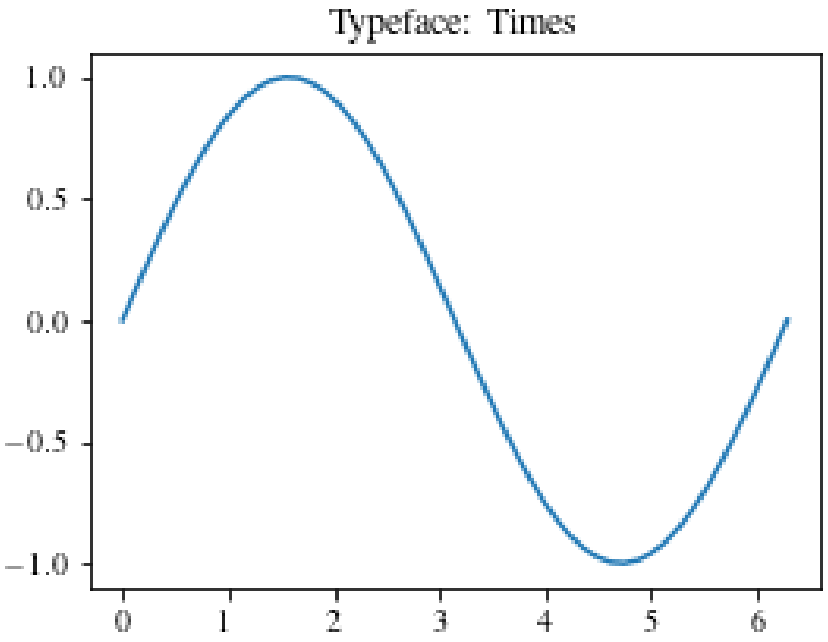
7.8 Workflow

1. Create a plot with `org-mode` babel with a python script.
   (a) `org-mode` is used for analysis, processing and presenting the results.
2. use the `:tangle` header argument to save the script into a centralized place.
   (a) If want to change a plot style, just run the script from this centralized place.
   (b) *<2021-09-20 Mon>* problem: where the data should be?

```
import numpy as np
import matplotlib.pyplot as plt
import figtex; figtex.style()

s = 300 / 72                    # 300 points in latex
fig, ax = plt.subplots(figsize=(s, s*3/4))
x = np.linspace(0, 2 * np.pi)
y = np.sin(x)
ax.plot(x, y)
ax.set_title(f"Typeface: {plt.rcParams['font.serif'][0]}")
```




```
import numpy as np
import matplotlib.pyplot as plt
import figtex; figtex.style(serif='Times')

s = 300 / 72                        # 300 points in latex
fig, ax = plt.subplots(figsize=(s, s*3/4))
x = np.linspace(0, 2 * np.pi)
y = np.sin(x)
ax.plot(x, y)
ax.set_title(f"Typeface:  {plt.rcParams['font.serif'][0]}")
```



Then I can run all scripts with,

```
python3 scripts/*
```

7.9 Plot resolution

7.9.1 Concepts

Matplotlib uses the concept of "dots per inches", dpi, to set the amount of pixels for the figure. The default dpi matplotlib uses is 100.

For the **size of figure**, matplotlib uses inches. The default figure size matplotlib uses is (6, 4) inches. So the total number of pixels (dots) is (600, 400).

Plot elements sizes:

1. lines, markers and others are created with "points" scale.
2. the relation between points and inches is 72 points per inches.
3. text is given in points, so a text size 10, will have 10/72 inches height.

7.9.2 Remarks

Remarks:

1. changing just figure size, keeps the plot elements with the same thickness (measured in points).

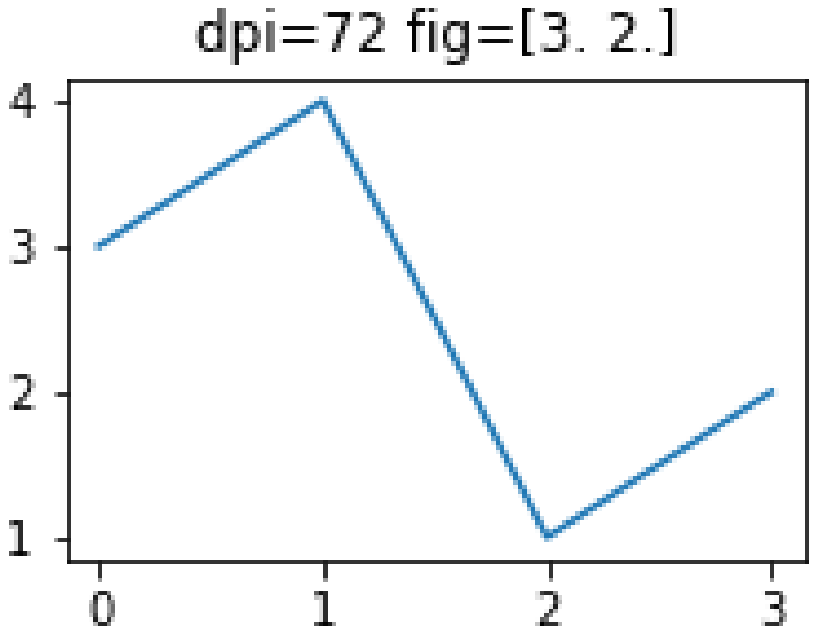
   (a) changing dpi, scales everything.

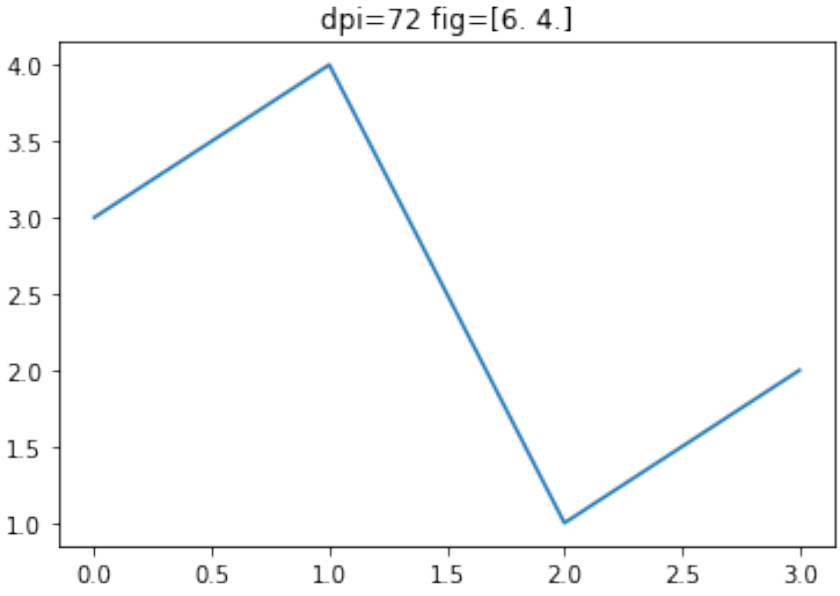
2. font size (in pt) does not change with figure size (in inches).

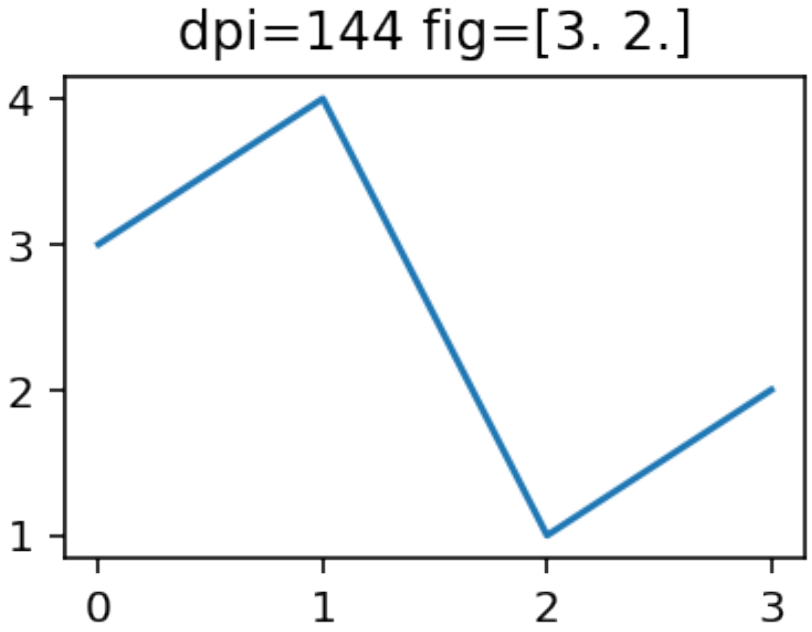
7.9.3 Tests

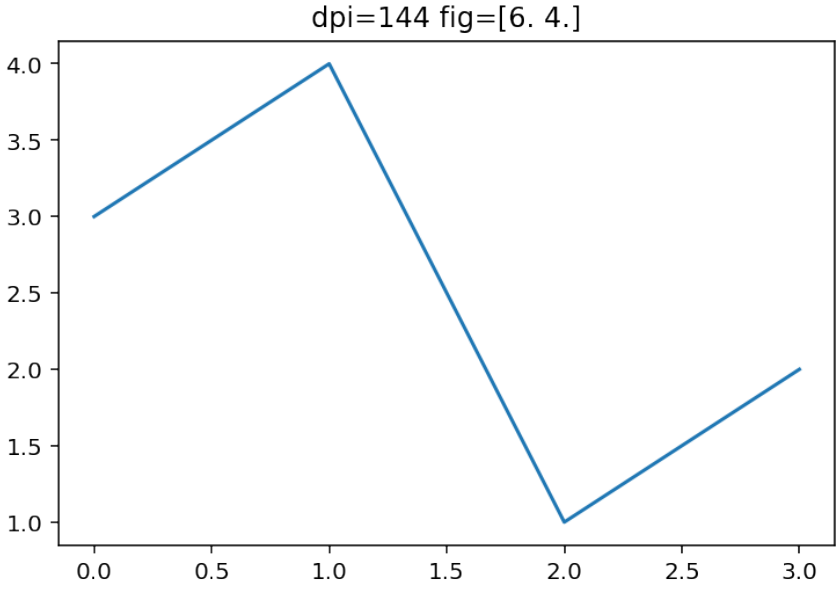
```
import matplotlib.pyplot as plt
def plot(figsize=(6, 4), dpi=72):
    fig, ax = plt.subplots(figsize=figsize, dpi=dpi)
    ax.plot([3, 4, 1, 2])
    ax.set_title(f"dpi={fig.get_dpi()} fig={fig.get_size_inches()}")
    fig.savefig(f'images/fig_dpi_{fig.get_dpi()}_size_{fig.get_size_inches()}.png')

print('default figsize', plt.rcParams['figure.figsize'])
print('default dpi', plt.rcParams['figure.dpi'])
plot((3, 2), 72)
plot((6, 4), 72)
plot((3, 2), 72 * 2)
plot((6, 4), 72 * 2)
```

```
default figsize [6.0, 4.0]
default dpi 72.0
```


dpi=72 fig=[3. 2.]
4
3
2
1
0
1
2
3

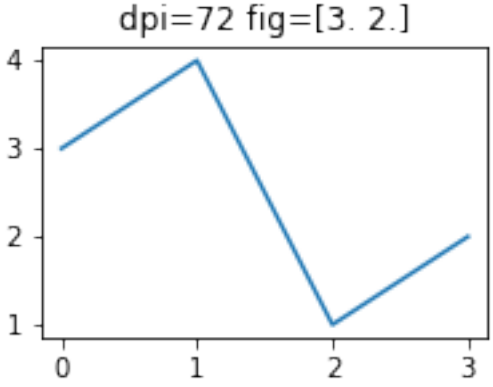

dpi=72 fig=[6. 4.]
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0

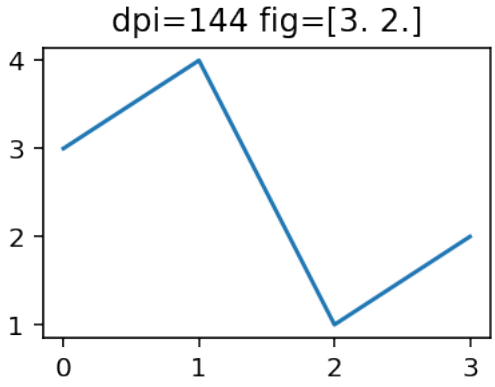

dpi=144 fig=[3. 2.]
4
3
2
1
0
1
2
3

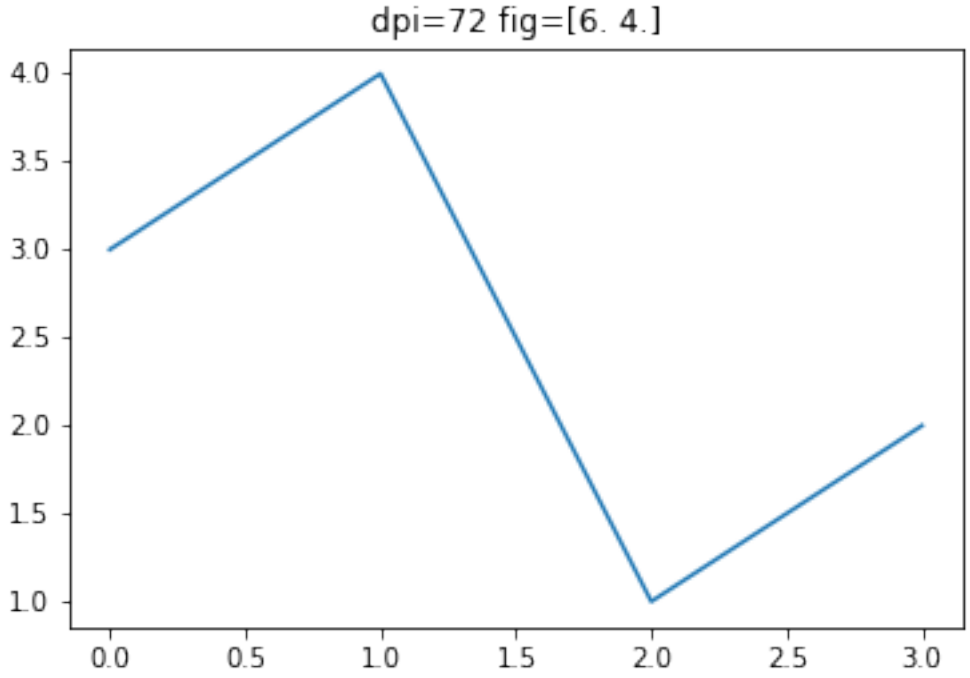

dpi=144 fig=[6. 4.]
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0

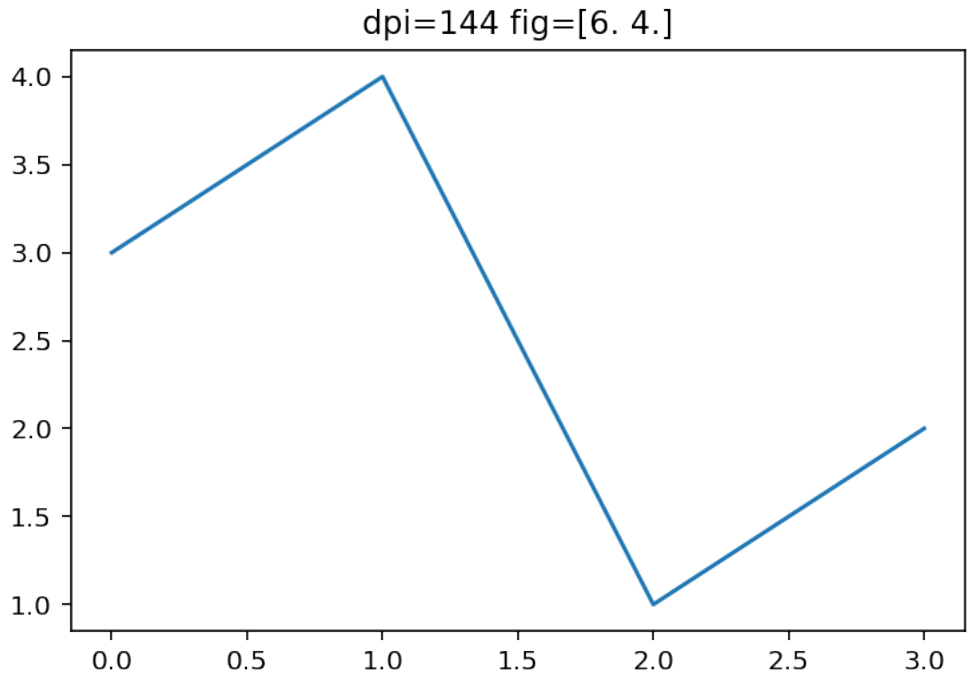
7.9.4 Relation with the font size

The font size is 11pt.








dpi=144 fig=[6. 4.]
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0